

CLEAN CLAIMS AFTER *RESPONSE A*

1 1. (FIRST AMENDED Per A) A computer-implemented method for dynamic emulation of legacy
2 instructions comprising:

3 accessing said legacy instructions,

4 for each particular legacy instruction,

5 translating the particular legacy instruction into one or more particular translated
6 instructions for emulating the particular legacy instruction,

7 if the particular legacy instruction is an operand-setting instruction for storing a value of
8 a precedent operand, setting a corresponding flag when the value of the precedent
9 operand has not been determined,

10 if the particular legacy instruction is an operand-using instruction for using the precedent
11 operand, checking to determine if the corresponding flag is set,

12 if the corresponding flag is set, suspending the translation of said operand-using
13 instruction,

14 executing said one or more particular translated instructions corresponding to
15 said operand-setting instruction to determine the value of said precedent
16 operand,

17 clearing the corresponding flag and resuming translation of said legacy instruc-
18 tions using the value of said precedent operand in the resumed translation,

19 if the corresponding flag is not set, continuing the translation of said operand-using
20 instruction,

21 continuing executing the translated instructions to emulate execution of the legacy instructions.

1 2. (Original) The method of Claim 1 wherein said step of suspending the translation inserts a
2 resume_translation call into a stream of said translated instructions in place of translations of said
3 operand-using instruction.

CLEAN CLAIMS AFTER *RESPONSE A*

- 1 3. (Original) The method of Claim 2 wherein said step of executing said one or more particular
2 translated instructions continues until said resume_translation call is reached and said step of
3 resuming translation resumes by translating said operand-using instruction.

- 1 4. (Original) The method of Claim 1 wherein said particular legacy instruction is an unpredictable
2 operand-using instruction having unpredictable byte alignment of operands.

- 1 5. (Original) The method of Claim 4 wherein said unpredictable operand-using instruction uses
2 operands of variable length.

- 1 6. (Original) The method of Claim 1 wherein if said if the particular legacy instruction is a
2 predictable operand-using instruction, said step of checking is bypassed to said step of continuing
3 the translation.

- 1 7. (Original) The method of Claim 6 wherein said predictable operand-using instruction employs
2 fixed-length operands.

- 1 8. (Original) The method of Claim 1 wherein said legacy instructions are for a legacy system
2 having a S/390 architecture.

- 1 9. (Original) The method of Claim 1 wherein said legacy instructions are object code instructions
2 compiled/assembled for a legacy architecture.

- 1 10. (Original) The method of Claim 1 wherein said translated instructions are for execution in a
2 RISC architecture.

CLEAN CLAIMS AFTER *RESPONSE A*

1 11. (FIRST AMENDED Per A) A computer-implemented method for dynamic emulation of legacy
2 instructions, where the legacy instructions are compiled/assembled into object code form for a native
3 architecture, where the legacy instructions are executed as guests in the host architecture, where the
4 legacy instructions are translated to translated instructions in the host architecture and the translated
5 instructions are executed in the host architecture concurrently with the translation of the legacy
6 instructions in the host architecture, comprising:

7 accessing said legacy instructions in a host system operating with said host architecture,
8 for each particular legacy instruction,

9 translating the particular legacy instruction into one or more particular translated
10 instructions of the host system for emulating the particular legacy instruction as a
11 guest in said host architecture,

12 if the particular legacy instruction is an operand-setting instruction for storing a value of
13 a precedent operand at execution time of said one or more particular translated
14 instructions, setting a corresponding flag in a flag store of said host system indicating
15 that the value of the precedent operand has not been determined,

16 if the particular legacy instruction is an operand-using instruction for using the precedent
17 operand, checking said flag store in said host system to determine if the correspond-
18 ing flag is set in said flag store,

19 if the corresponding flag is set in said flag store, suspending the translation in
20 said host system of said operand-using instruction,

21 executing in said host system said one or more particular translated
22 instructions corresponding to said operand-setting instruction to
23 determine the value of said precedent operand,

24 initializing all flags to zero in said flag store and resuming translation of
25 said legacy instructions in said host system using the value of said
26 precedent operand in the resumed translation,

27 if the corresponding flag in said flag store is not set, continuing the translation of
28 said operand-using instruction in said host system,

CLEAN CLAIMS AFTER *RESPONSE A*

29 continuing executing the translated instructions to emulate execution of the legacy instructions.

1 12. (FIRST AMENDED Per A) A computer system for dynamic emulation of legacy instructions
2 comprising:

3 an access unit for accessing said legacy instructions,

4 a flag store for storing flags to indicate when the values of precedent operands have not been
5 determined,

6 an execution unit for executing translated instructions to emulate execution of the legacy
7 instructions

8 a legacy code translator operating for each particular legacy instruction,

9 to translate the particular legacy instruction into one or more particular translated
10 instructions for emulating the particular legacy instruction,

11 if the particular legacy instruction is an operand-setting instruction for storing a value of
12 a precedent operand, to set a corresponding flag in said flag store when the value of
13 the precedent operand has not been determined,

14 if the particular legacy instruction is an operand-using instruction for using the precedent
15 operand, to check said flag store to determine if the corresponding flag is set,

16 if the corresponding flag is set in said flag store, to suspend the translation of
17 said operand-using instruction, and

18 to execute said one or more particular translated instructions
19 corresponding to said operand-setting instruction to determine
20 the value of said precedent operand,

21 to initialize all flags to zero in said flag store and resume transla-
22 tion of said legacy instructions using the value of said
23 precedent operand in the resumed translation,

24 if the corresponding flag is not set, to continue the translation of said
25 operand-using instruction.

CLEAN CLAIMS AFTER *RESPONSE A*

- 1 13. (Original) The system of Claim 12 wherein said legacy code translator in suspending the
2 translation inserts a `resume_translation` call into a stream of said translated instructions in place of
3 translations of said operand-using instruction.

- 1 14. (Original) The system of Claim 13 wherein said legacy code translator in executing said one
2 or more particular translated instructions continues until said `resume_translation` call is reached and
3 resumes translation by translating said operand-using instruction.

- 1 15. (Original) The system of Claim 12 wherein said particular legacy instruction is an unpredict-
2 able operand-using instruction having unpredictable byte alignment of operands.

- 1 16. (Original) The system of Claim 15 wherein said unpredictable operand-using instruction uses
2 operands of variable length.

- 1 17. (Original) The system of Claim 12 wherein if said if the particular legacy instruction is a
2 predictable operand-using instruction, said legacy code translator operates to bypass the checking
3 of said flag store and continues the translation of said legacy instructions.

- 1 18. (Original) The system of Claim 17 wherein said predictable operand-using instruction employs
2 fixed-length operands.

- 1 19. (Original) The system of Claim 12 wherein said legacy instructions are for a legacy system
2 having a S/390 architecture.

- 1 20. (Original) The system of Claim 12 wherein said legacy instructions are object code instructions
2 compiled/assembled for a legacy architecture.

CLEAN CLAIMS AFTER *RESPONSE A*

1 21. (Original) The system of Claim 12 wherein said translated instructions are for execution in a
2 RISC architecture.

INTERLINED CLAIMS AFTER *RESPONSE A*

- 1 1. (FIRST AMENDED Per A) A computer-implemented method for dynamic emulation of legacy
2 instructions comprising:
3 accessing said legacy instructions,
4 for each particular legacy instruction,
5 translating the particular legacy instruction into one or more particular translated
6 instructions for emulating the particular legacy instruction,
7 if the particular legacy instruction is an operand-setting instruction for storing a value of
8 a precedent operand, setting a corresponding flag when the value of the precedent
9 operand has not been determined,
10 if the particular legacy instruction is an operand-using instruction for using the precedent
11 operand, checking to determine if the corresponding flag is set,
12 if the corresponding flag is set, suspending the translation of said operand-using
13 instruction,
14 executing said one or more particular translated instructions corresponding to
15 said operand-setting instruction to determine the value of said precedent
16 operand,
17 clearing the corresponding flag and resuming translation of said legacy instruc-
18 tions using the value of said precedent operand in the resumed translation,
19 if the corresponding flag is not set, continuing the translation of said operand-using
20 instruction,
21 continuing executing the translated instructions to emulate execution of the legacy instructions.

- 1 2. (Original) The method of Claim 1 wherein said step of suspending the translation inserts a
2 resume_translation call into a stream of said translated instructions in place of translations of said
3 operand-using instruction.

INTERLINED CLAIMS AFTER *RESPONSE A*

- 1 3. (Original) The method of Claim 2 wherein said step of executing said one or more particular
2 translated instructions continues until said resume_translation call is reached and said step of
3 resuming translation resumes by translating said operand-using instruction.

- 1 4. (Original) The method of Claim 1 wherein said particular legacy instruction is an unpredictable
2 operand-using instruction having unpredictable byte alignment of operands.

- 1 5. (Original) The method of Claim 4 wherein said unpredictable operand-using instruction uses
2 operands of variable length.

- 1 6. (Original) The method of Claim 1 wherein if said if the particular legacy instruction is a
2 predictable operand-using instruction, said step of checking is bypassed to said step of continuing
3 the translation.

- 1 7. (Original) The method of Claim 6 wherein said predictable operand-using instruction employs
2 fixed-length operands.

- 1 8. (Original) The method of Claim 1 wherein said legacy instructions are for a legacy system
2 having a S/390 architecture.

- 1 9. (Original) The method of Claim 1 wherein said legacy instructions are object code instructions
2 compiled/assembled for a legacy architecture.

- 1 10. (Original) The method of Claim 1 wherein said translated instructions are for execution in a
2 RISC architecture.

INTERLINED CLAIMS AFTER *RESPONSE A*

1 11. (FIRST AMENDED Per A) A computer-implemented method for dynamic emulation of legacy
2 instructions, where the legacy instructions are compiled/assembled into object code form for a native
3 architecture, where the legacy instructions are executed as guests in the host architecture, where the
4 legacy instructions are translated to translated instructions in the host architecture and the translated
5 instructions are executed in the host architecture concurrently with the translation of the legacy
6 instructions in the host architecture, comprising:

7 accessing said legacy instructions in a host system operating with said host architecture,
8 for each particular legacy instruction,

9 translating the particular legacy instruction into one or more particular translated
10 instructions of the host system for emulating the particular legacy instruction as a
11 guest in said host architecture,

12 if the particular legacy instruction is an operand-setting instruction for storing a value of
13 a precedent operand at execution time of said one or more particular translated
14 instructions, setting a corresponding flag in a flag store of said host system indicating
15 that the value of the precedent operand has not been determined,

16 if the particular legacy instruction is an operand-using instruction for using the precedent
17 operand, checking said flag store in said host system to determine if the correspond-
18 ing flag is set in said flag store,

19 if the corresponding flag is set in said flag store, suspending the translation in
20 said host system of said operand-using instruction,

21 executing in said host system said one or more particular translated
22 instructions corresponding to said operand-setting instruction to
23 determine the value of said precedent operand,

24 initializing all flags to zero in said flag store and resuming translation of
25 said legacy instructions in said host system using the value of said
26 precedent operand in the resumed translation,

27 if the corresponding flag in said flag store is not set, continuing the translation of
28 said operand-using instruction in said host system.

INTERLINED CLAIMS AFTER *RESPONSE A*

29 continuing executing the translated instructions to emulate execution of the legacy instructions.

1 12. (FIRST AMENDED Per A) A computer system for dynamic emulation of legacy instructions
2 comprising:

3 an access unit for accessing said legacy instructions,

4 a flag store for storing flags to indicate when the values of precedent operands have not been
5 determined,-

6 an execution unit for executing translated instructions to emulate execution of the legacy
7 instructions.

8 a legacy code translator operating for each particular legacy instruction,

9 to translate the particular legacy instruction into one or more particular translated
10 instructions for emulating the particular legacy instruction,

11 if the particular legacy instruction is an operand-setting instruction for storing a value of
12 a precedent operand, to set a corresponding flag in said flag store when the value of
13 the precedent operand has not been determined,

14 if the particular legacy instruction is an operand-using instruction for using the precedent
15 operand, to check said flag store to determine if the corresponding flag is set,

16 if the corresponding flag is set in said flag store, to suspend the translation of
17 said operand-using instruction, and

18 to execute said one or more particular translated instructions
19 corresponding to said operand-setting instruction to determine
20 the value of said precedent operand,

21 to initialize all flags to zero in said flag store and resume transla-
22 tion of said legacy instructions using the value of said
23 precedent operand in the resumed translation,

24 if the corresponding flag is not set, to continue the translation of said
25 operand-using instruction.

INTERLINED CLAIMS AFTER *RESPONSE A*

- 1 13. (Original) The system of Claim 12 wherein said legacy code translator in suspending the
2 translation inserts a resume_translation call into a stream of said translated instructions in place of
3 translations of said operand-using instruction.

- 1 14. (Original) The system of Claim 13 wherein said legacy code translator in executing said one
2 or more particular translated instructions continues until said resume_translation call is reached and
3 resumes translation by translating said operand-using instruction.

- 1 15. (Original) The system of Claim 12 wherein said particular legacy instruction is an
2 unpredictable operand-using instruction having unpredictable byte alignment of operands.

- 1 16. (Original) The system of Claim 15 wherein said unpredictable operand-using instruction uses
2 operands of variable length.

- 1 17. (Original) The system of Claim 12 wherein if said if the particular legacy instruction is a
2 predictable operand-using instruction, said legacy code translator operates to bypass the checking
3 of said flag store and continues the translation of said legacy instructions.

- 1 18. (Original) The system of Claim 17 wherein said predictable operand-using instruction employs
2 fixed-length operands.

- 1 19. (Original) The system of Claim 12 wherein said legacy instructions are for a legacy system
2 having a S/390 architecture.

- 1 20. (Original) The system of Claim 12 wherein said legacy instructions are object code
2 instructions compiled/assembled for a legacy architecture.

INTERLINED CLAIMS AFTER *RESPONSE A*

1 21. (Original) The system of Claim 12 wherein said translated instructions are for execution in a
2 RISC architecture.